

EQUITO TESLA

Pregunta 1:

Obtén la lista de personajes que están vivos (status: "Alive"). Limita los resultados a 5 personajes y alfabéticamente por nombre.

```
1 [
2   {
3     $match:
4     /**
5      * query: The query in MQL.
6      */
7     {
8       status: "Alive",
9     },
10  },
11  {
12    $limit: 5,
13  },
14  {
15    $sort: {
16      name: 1,
17    },
18  },
19 ]
```

olivirus.personajes 120 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Pipeline **\$match** \$limit \$sort Generate aggregation Explain Export Run More Options

Untitled - modified **SAVE** **CREATE NEW** **EXPORT TO LANGUAGE** **PREVIEW** **STAGES** **TEXT**

Stage 1 (\$match) Output after \$match stage (Sample of 5 documents)

_id	status	name	species	type	gender	origin	location
ObjectId('6529a376c881eff9476295d8')	Alive	Rick Sanchez	Human	Male	Human	Object	Object
ObjectId('6529a376c881eff9476295d9')	Alive	Morty Smith	Human	Male	Human	Object	Object
ObjectId('6529a376c881eff9476295da')	Alive	Summer Smith	Human	Female	Human	Object	Object
ObjectId('6529a376c881eff9476295db')	Alive	Beth Smith	Human	Female	Human	Object	Object
ObjectId('6529a376c881eff9476295dc')	Alive	Jerry Smith	Human	Male	Human	Object	Object

Stage 2 (\$limit) Output after \$limit stage (Sample of 5 documents)

_id	status	name	species	type	gender	origin	location
ObjectId('6529a376c881eff9476295d8')	Alive	Rick Sanchez	Human	Male	Human	Object	Object
ObjectId('6529a376c881eff9476295d9')	Alive	Morty Smith	Human	Male	Human	Object	Object
ObjectId('6529a376c881eff9476295da')	Alive	Summer Smith	Human	Female	Human	Object	Object
ObjectId('6529a376c881eff9476295db')	Alive	Beth Smith	Human	Female	Human	Object	Object
ObjectId('6529a376c881eff9476295dc')	Alive	Jerry Smith	Human	Male	Human	Object	Object

Stage 3 (\$sort) Output after \$sort stage (Sample of 5 documents)

_id	status	name	species	type	gender	origin	location
ObjectId('6529a376c881eff9476295da')	Alive	Summer Smith	Human	Female	Human	Object	Object
ObjectId('6529a376c881eff9476295d9')	Alive	Morty Smith	Human	Male	Human	Object	Object
ObjectId('6529a376c881eff9476295d8')	Alive	Rick Sanchez	Human	Male	Human	Object	Object
ObjectId('6529a376c881eff9476295db')	Alive	Beth Smith	Human	Female	Human	Object	Object
ObjectId('6529a376c881eff9476295dc')	Alive	Jerry Smith	Human	Male	Human	Object	Object

Shell:

```
db.personajes.aggregate([{$match: {status: "Alive"}}, {$limit: 5}, {$sort: {name: 1}}]);
```

Pregunta 2:

Encuentra el número total de personajes de género masculino (gender: "Male").

olivirus.personajes

Documents Aggregations Schema Indexes Validation

120 DOCUMENTS 1 INDEXES

Pipeline **Match** **Count** Generate aggregation Explain Export Run More Options

Untitled - modified SAVE CHLATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT

Stage 1 **Match**

```
1 // **
2 * query: The query in MQL.
3 */
4 {
5   gender: "Male",
6 }
```

Output after **Match** stage (Sample of 10 documents)

```
{
  "_id": "ObjectID('6529e376c88e9f9476295de')",
  "id": 2,
  "name": "Morty Smith",
  "status": "Alive",
  "species": "Human",
  "type": "",
  "gender": "Male",
  "origin": Object,
  "location": Object
}
```

Stage 2 **Count**

```
1 // **
2 * Provide the field name for the count.
3 */
4 "count"
```

Output after **Count** stage (Sample of 1 document)

```
{
  "count": 82
}
```

Shell:

```
db.personajes.aggregate([{$match: {gender: "Male"}}, {$count: "count"}]);
```

```
Type "it" for more
olivirus> db.personajes.aggregate([{$match: {gender: "Male"}}, {$count: "count"}]);
[ { count: 82 } ]
```

Pregunta 3:

Obtén una lista de las especies únicas presentes en la base de datos.

```
▼ Stage1 $group ☒
1 ▼ /**
2   * _id: The id of the group.
3   * fieldN: The first field name.
4   */
5 ▼ {
6   _id: "$species",
7   cantidad: {
8     $sum: 1,
9   },
10 }
```

PIPELINE OUTPUT		OUTPUT OPTIONS ▼
Sample of 10 documents		
_id: "Alien"	cantidad: 30	
_id: "Mythological Creature"	cantidad: 6	
_id: "Human"	cantidad: 61	
_id: "Disease"	cantidad: 6	
_id: "Animal"	cantidad: 2	
_id: "Humanoid"	cantidad: 8	
_id: "Poopybutthole"	cantidad: 2	
_id: "Robot"	cantidad: 2	
_id: "unknown"	cantidad: 1	

Shell:

```
db.personajes.aggregate([{$group: {_id: "$species", cantidad: {$sum: 1}}}]
```

```
olivirus> db.personajes.aggregate([{$group: {_id: "$species", cantidad: {$sum: 1}}}]
[
  { _id: 'Cronenberg', cantidad: 2 },
  { _id: 'Mythological Creature', cantidad: 6 },
  { _id: 'Poopybutthole', cantidad: 2 },
  { _id: 'Robot', cantidad: 2 },
  { _id: 'unknown', cantidad: 1 },
  { _id: 'Animal', cantidad: 2 },
  { _id: 'Humanoid', cantidad: 8 },
  { _id: 'Alien', cantidad: 30 },
  { _id: 'Human', cantidad: 61 },
  { _id: 'Disease', cantidad: 6 }
]
```

Pregunta 4:

Encuentra el número de episodios en los que aparece el personaje con ID 6.

Shell:

```
db.personajes.findOne({id: 6}).episode.length)
```

```
olivirus> db.personajes.findOne({id: 6}).episode.length
1
olivirus> print(db.personajes.findOne({id: 6}).episode.length)
1
```

```
db.personajes.findOne({id: 1}).episode.length)
```

```
olivirus> print(db.personajes.findOne({id: 1}).episode.length)
51
```

PIPELINE OUTPUT

Sample of 10 documents

```
_id: ObjectId('6529e376c081eff9476295dd')
id: 1
name: "Rick Sanchez"
status: "Alive"
species: "Human"
type: ""
gender: "Male"
▶ origin: Object
▶ location: Object
image: "https://rickandmortyapi.com/api/character/avatar/1.jpeg"
▶ episode: Array (51)
url: "https://rickandmortyapi.com/api/character/1"
created: "2017-11-04T18:48:46.250Z"
```

Pregunta 5:

Encuentra el promedio de episodios en los que aparecen los personajes de género femenino.

```
1 ▾ [
2 ▾ {
3 ▾   $match: {
4     gender: "Male",
5   },
6   },
7   {
8     $group: {
9       _id: null,
10      // Agrupamos todos porque queremos un promedio general
11      averageEpisodes: {
12        $avg: {
13          $size: "$episode",
14        },
15      },
16    },
17  },
18 ]
```

PIPELINE OUTPUT

Sample of 1 document

_id: null

averageEpisodes: 3.1097560975609757

Shell:

```
db.personajes.aggregate([{$match: {gender: "Male"}}, {$group: {_id: null, averageEpisodes:
{$avg: {$size: "$episode"}}}}]);
```

```
olivirus> db.personajes.aggregate([{$match: {gender: "Male"}}, {$group: {_id: null, averageEpisodes: {$avg: {$size: "$episode"}}}}]);
[ { _id: null, averageEpisodes: 3.1097560975609757 } ]
olivirus> |
```

Pregunta 6:

Agrupar los personajes por su estado ("status") y muestra cuántos personajes están vivos y cuántos están muertos en cada grupo.

The screenshot shows the MongoDB Atlas query editor with two stages. Stage 1 is a \$match query filtering documents where status is either 'Alive' or 'Dead'. Stage 2 is a \$group query grouping documents by status and counting them.

Stage 1: \$match

```
1 // **
2 // query: The query in MQL.
3 //
4 {
5   status: {
6     $in: ["Alive", "Dead"],
7   },
8 }
```

Output after \$match stage (Sample of 10 documents)

Document 1	Document 2	Document 3
{ "_id": ObjectId("6529e376c881eff9476295de"), "id": 1, "name": "Rick Sanchez", "status": "Alive", "species": "Human", "type": "", "gender": "Male", "origin": Object, "location": Object }	{ "_id": ObjectId("6529e376c881eff9476295de"), "id": 2, "name": "Morty Smith", "status": "Alive", "species": "Human", "type": "", "gender": "Male", "origin": Object, "location": Object }	{ "_id": ObjectId("6529e376c881eff9476295df"), "id": 3, "name": "Summer Smith", "status": "Alive", "species": "Human", "type": "", "gender": "Female", "origin": Object, "location": Object }

Stage 2: \$group

```
1 // **
2 // _id: The id of the group.
3 // field: The first field name.
4 //
5 {
6   _id: "$status",
7   sum: {
8     $sum: 1,
9   },
10 }
```

Output after \$group stage (Sample of 2 documents)

Group 1	Group 2
{ "_id": "Alive", "sum": 51 }	{ "_id": "Dead", "sum": 50 }

Shell:

```
db.personajes.aggregate([{$match: {status: {$in: ["Alive", "Dead"]}}}, {$group: {_id: "$status", sum: {$sum: 1}}}]
```

```
olivirus> db.personajes.aggregate([{$match: {status: {$in: ["Alive", "Dead"]}}}, {$group: {_id: "$status", sum: {$sum: 1}}}]
[ { _id: 'Dead', sum: 50 }, { _id: 'Alive', sum: 51 } ]
olivirus>
```

Pregunta 7:

Encuentra el número de episodios en los que aparece cada personaje. Muestra el resultado con el nombre del personaje y la cantidad de episodios en los que aparece.

The screenshot shows the MongoDB Atlas Aggregations Builder interface. The top bar indicates 120 documents and 1 index. The 'Pipeline' tab is active, showing a single stage named '\$project'. Below the pipeline, a 'Preview of documents' section displays four sample documents. The first document is for 'Rick Sanchez' with 51 episodes. The second is for 'Morty Smith' with 51 episodes. The third is for 'Summer Smith' with 42 episodes. The fourth is for 'Beth Smith' with 42 episodes. The 'Stage 1 \$project' configuration is shown on the left, and the 'Output after \$project stage' is shown on the right, displaying the same four documents with the 'episodios' field added.

Shell:

```
olivirus> db.personajes.aggregate([{$project: {_id: 0, name: 1, episodios: {$size: "$episodios"}}}]);
```

```
olivirus> db.personajes.aggregate([{$project: {_id: 0, name: 1, episodios: {$size: "$episodios"}}}]);
[
  { name: 'Rick Sanchez', episodios: 51 },
  { name: 'Morty Smith', episodios: 51 },
  { name: 'Summer Smith', episodios: 42 },
  { name: 'Beth Smith', episodios: 42 },
  { name: 'Jerry Smith', episodios: 39 },
  { name: 'Abadango Cluster Princess', episodios: 1 },
  { name: 'Abradolf Lincler', episodios: 2 },
  { name: 'Adjudicator Rick', episodios: 1 },
  { name: 'Agency Director', episodios: 1 },
  { name: 'Alan Rails', episodios: 1 },
  { name: 'Albert Einstein', episodios: 1 },
  { name: 'Alexander', episodios: 1 },
  { name: 'Alien Googah', episodios: 1 },
  { name: 'Alien Morty', episodios: 1 },
  { name: 'Alien Rick', episodios: 1 },
  { name: 'Amish Cyborg', episodios: 1 },
  { name: 'Annie', episodios: 1 },
  { name: 'Antenna Morty', episodios: 2 },
  { name: 'Antenna Rick', episodios: 1 },
  { name: 'Ants in my Eyes Johnson', episodios: 1 }
]
```

Cómo existen libros repetidos, podemos:

Pipeline
\$group
\$project
Generate aggregation
Explain
Export
Run

Untitled - modified
SAVE
CREATE NEW
EXPORT TO LANGUAGE
PREVIEW
STAGES

Stage 1 \$group
Output after \$group stage (Sample of 10 documents)

```

1 {
2   _id: "$name",
3
4   episodios: {
5     $first: {
6       $size: "$episodio",
7     },
8   },
9 }

```

```

_id: "Blue Footprint Guy"
episodios: 1

_id: "Armothy"
episodios: 1

_id: "Alien Rick"
episodios: 1

```

Stage 2 \$project
Output after \$project stage (Sample of 10 documents)

```

1 {
2   _id: 0,
3   name: "$_id",
4   episodios: 1,
5 }

```

```

episodios: 1
name: "Blue Footprint Guy"

episodios: 1
name: "Armothy"

episodios: 1
name: "Alien Rick"

```

Shell: db.personajes.aggregate([{\$group: {_id: "\$name", episodios: {\$first: {\$size: "\$episodio"}}}}, {\$project: {_id: 0, name: "\$_id", episodios: 1}}]);

```

olivirus> db.personajes.aggregate([{$group: {_id: "$name", episodios: {$first: {$size: "$episodio"}}}}, {$project: {_id: 0, name: "$_id", episodios: 1}}]);
[
  { episodios: 1, name: 'Blue Footprint Guy' },
  { episodios: 1, name: 'Armothy' },
  { episodios: 1, name: 'Alien Rick' },
  { episodios: 1, name: 'Cop Rick' },
  { episodios: 1, name: 'Calypso' },
  { episodios: 42, name: 'Summer Smith' },
  { episodios: 2, name: 'Boobloosian' },
  { episodios: 2, name: 'Diane Sanchez' },
  { episodios: 1, name: 'Dr. Xenon Bloom' },
  { episodios: 1, name: 'Antenna Rick' },
  { episodios: 1, name: 'Diablo Verde' },
  { episodios: 3, name: 'Benjamin' },
  { episodios: 1, name: 'Bill' },
  { episodios: 1, name: 'Eli's Girlfriend' },
  { episodios: 1, name: 'E. Coli' },
  { episodios: 1, name: 'Doom-Nomitron' },
  { episodios: 1, name: 'Blue Shirt Morty' },
  { episodios: 1, name: 'Alien Morty' },
  { episodios: 1, name: 'Armageddon' },
  { episodios: 7, name: 'Birdperson' }
]

```


Pregunta 8:

Encuentra la ubicación más común de los personajes. Muestra el nombre de la ubicación y cuántos personajes están allí.

olivirus.personajes 120 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Pipeline ▾ \$group \$sort \$limit Generate aggregation ⚙ Explain Export Run More Options ▶

Untitled - modified SAVE + CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT

120 Documents in the collection

Preview of documents

<pre>{ "_id": "Rick Sanchez", "status": "Alive", "species": "Human", "type": "Male", "gender": "Male", "origin": "Earth (Replacement Dimension)" }</pre>	<pre>{ "_id": "Morty Smith", "status": "Alive", "species": "Human", "type": "Male", "gender": "Male", "origin": "Earth (Replacement Dimension)" }</pre>	<pre>{ "_id": "Summer Smith", "status": "Alive", "species": "Human", "type": "Female", "gender": "Female", "origin": "Earth (C-500A)" }</pre>	<pre>{ "_id": "Beth Smith", "status": "Alive", "species": "Human", "type": "Female", "gender": "Female", "origin": "Earth (C-137)" }</pre>
--	---	---	--

Stage 1 \$group

```
1 {
2   "_id": "$location.name",
3   "count": {
4     "$sum": 1,
5   },
6 }
```

Output after \$group stage (Sample of 10 documents)

<pre>{ "_id": "Earth (Replacement Dimension)", "count": 21 }</pre>	<pre>{ "_id": "Earth (C-500A)", "count": 2 }</pre>	<pre>{ "_id": "Planet Squanch", "count": 3 }</pre>
--	--	--

Stage 2 \$sort

```
1 {
2   "count": -1,
3 }
```

Output after \$sort stage (Sample of 10 documents)

<pre>{ "_id": "Citadel of Ricks", "count": 27 }</pre>	<pre>{ "_id": "Earth (Replacement Dimension)", "count": 21 }</pre>	<pre>{ "_id": "Earth (C-137)", "count": 11 }</pre>
---	--	--

Stage 3 \$limit

```
1 1
```

Output after \$limit stage (Sample of 1 document)

```
{
  "_id": "Citadel of Ricks",
  "count": 27
}
```

Shell:

```
olivirus> db.personajes.aggregate([{$group: {_id: "$location.name", count: {$sum: 1}}}, {$sort: {count: -1}}, {$limit: 1}]);
[ { _id: 'Citadel of Ricks', count: 27 } ]
```

Pregunta 9:

Proyecta una lista de personajes que incluya solo sus nombres, género y estado. Limita los resultados a 3 personajes y ordénalos alfabéticamente por nombre.

olivirus.personajes 120 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Pipeline ▾ \$project \$limit \$sort Generate aggregation ⓘ Explain Export Run More Options ▸

Untitled - modified SAVE CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT

Stage 1 \$project

```
1 /**
2  * specifications: The fields to
3  * include or exclude.
4  */
5 {
6   _id:0,
7   name:1,
8   gender:1,
9   status:1
10 }
11
```

Output after \$project stage (Sample of 10 documents)

name: "Rick Sanchez" status: "Alive" gender: "Male"	name: "Morty Smith" status: "Alive" gender: "Male"	name: "Summer Smith" status: "Alive" gender: "Female"
---	--	---

Stage 2 \$limit

```
1 /**
2  * Provide the number of documents to limit
3  */
4 3
```

Output after \$limit stage (Sample of 3 documents)

name: "Rick Sanchez" status: "Alive" gender: "Male"	name: "Morty Smith" status: "Alive" gender: "Male"	name: "Summer Smith" status: "Alive" gender: "Female"
---	--	---

Stage 3 \$sort

```
1 /**
2  * Provide any number of field/order pair
3  */
4 {
5   name: 1
6 }
```

Output after \$sort stage (Sample of 3 documents)

name: "Morty Smith" status: "Alive" gender: "Male"	name: "Rick Sanchez" status: "Alive" gender: "Male"	name: "Summer Smith" status: "Alive" gender: "Female"
--	---	---

Shell: `db.personajes.aggregate([{$project: {_id: 0, name: 1, gender: 1, status: 1}}, {$limit: 3}, {$sort: {name: 1}}]);`

```
olivirus> db.personajes.aggregate([{$project: {_id: 0, name: 1, gender: 1, status: 1}}, {$limit: 3}, {$sort: {name: 1}}]);
[
  { name: 'Morty Smith', status: 'Alive', gender: 'Male' },
  { name: 'Rick Sanchez', status: 'Alive', gender: 'Male' },
  { name: 'Summer Smith', status: 'Alive', gender: 'Female' }
]
```